WHAT IS CLAIMED IS:

1. A communication network router containing an egress queuing structure, said queuing structure comprising:

a plurality of substantially parallel queues, each said queue having an input terminal and an output terminal;

a common shared memory device interconnected with all of said plurality of queues;

a queue congestion processor interconnected with said input terminals of all of said queues;

an output selection block interconnected with said output terminals of all of said queues; and

an egress arbitration processor interconnected with said output selection block.

2. The router of claim 1 wherein said queue congestion processor is interconnected with an egress reassembly logic.

3. The router of claim 1 wherein said output selection block is interconnected with a single tributary of a router egress port.

4. The router of claim 1 wherein each said queue is assigned to a single quality of service (QOS) priority level.

5. The router of claim 4 wherein each said queue is assigned to a different QOS priority level.

6. The router of claim 5 wherein said plurality of queues are assigned to four different said QOS priority levels.

7. The router of claim 1 further comprising a plurality of said egress queuing structures, each said queuing structure comprising:

a plurality of substantially parallel queues, each said queue having an input terminal and an output terminal;

a common shared memory device interconnected with all of said plurality of queues;

a queue congestion processor interconnected with said input terminals of all of said queues;

an output selection block interconnected with said output terminals of all of said queues; and

an egress arbitration processor interconnected with said output selection block.

8. The router of claim 7 wherein said queue congestion processor is interconnected with an egress reassembly logic.

9. The router of claim 7 wherein said output selection block of each said egress queuing structure is interconnected with a single tributary of a router egress port.

10. The router of claim 9 wherein a plurality of said tributaries are interconnected with a single router egress port.

11. A method of egress queue management in a router system comprising:

receiving packets having payload pointers, packet data payloads, and packet lengths into a queuing structure comprising a plurality of queues;

assigning said packets to separate queues in accordance with their quality of service (QOS) priority levels;

storing said packet payload pointers in said queues;

storing said packet payloads in a common memory pool shared by all of said plurality of queues; and

releasing said packets from said queues into a common egress tributary using a rate metering mechanism.

12. The method of claim 11 wherein said separate queues are assigned among four different said QOS priority levels.

13. The method of claim 11 wherein said rate metering mechanism comprises:

pre-allocating a bandwidth for each QOS priority level;

periodically adding tokens to a counter associated with each said QOS priority level queue, such that said tokens are added to each said counter at a time averaged rate substantially proportional to said pre-allocated bandwidth of said QOS priority level; and

applying a rate metering algorithm to said queues.

14. The method of claim 13 further comprising setting maximum and minimum limits on the number of said tokens in said counter.

15. The method of claim 13 wherein said rate metering algorithm is implemented in hardware.

16. The method of claim 13 wherein said rate metering algorithm comprises:

for each QOS priority level queue in sequence, starting with the highest QOS priority level, if there is a packet in said queue and if there are positive tokens in the counter associated with said queue, then releasing a packet from said queue; otherwise

for each QOS priority level queue in sequence, starting with the highest QOS priority level, if there is a packet in said queue, then releasing a packet from said queue regardless of whether tokens are in said counter; and

for each said packet released from a queue, deducting a number of tokens from the counter associated with said queue in proportion to the size of said packet.

17. The method of claim 11 further comprising instantaneous queue congestion management of drop probabilities of said packets using a queue congestion management algorithm before assigning said packets to said queues .

18. The method of claim 17 wherein said queue congestion management algorithm is implemented in hardware.

19. The method of claim 17 wherein said queue congestion management algorithm uses a floating point format.

20. The method of claim 19 wherein said floating point format comprises:

a four-bit normalized mantissa;

a six-bit biased exponent; and

performing multiply and divide operations with a mantissa table lookup.

21. The method of claim 20 wherein the most significant bit of said mantissa is implied.

22. The method of claim 20 wherein said biased exponent is in a range of -32 to +31.

23. The method of claim 20 wherein said mantissa table lookup uses two three-bit inputs and generates one four-bit output.

24. The method of claim 20 wherein said floating point format comprises only positive numbers.

25. The method of claim 17 wherein said queue congestion management algorithm comprises:

determining the total amount of shared memory space in bytes;

monitoring the instantaneous actual sizes of each of said queues;

dynamically calculating minimum and maximum queue sizes of a drop probability curve for each of said queues;

comparing said instantaneous actual sizes with said maximum and said minimum queue sizes;

if said instantaneous queue size is less than said minimum queue size, then assigning said packet to said queue; otherwise

if said instantaneous queue size is between said maximum and said minimum queue sizes, then calculating and applying a drop probability using the slope of said drop probability curve; and otherwise

if said instantaneous queue size is greater than said maximum queue size, then dropping said packet.

26. The method of claim 17 wherein non-utilized shared memory space is allocated simultaneously to all of said queues sharing said common memory pool.

27. The method of claim 11 further comprising time averaged congestion management of drop probabilities of said packets before assigning said packets to said queues using a weighted random early discard (WRED) algorithm applied to said queues having different QOS priority levels and sharing a common memory pool.

28. The method of claim 27 wherein said WRED algorithm is implemented in hardware.

29. The method of claim 27 wherein said WRED algorithm uses a floating point format.

30. The method of claim 29 wherein said floating point format comprises:

a four-bit normalized mantissa;

a six-bit biased exponent; and

performing multiply and divide operations with a mantissa table lookup.

31. The method of claim 30 wherein the most significant bit of said mantissa is implied.

32. The method of claim 30 wherein said biased exponent is in a range from -32 to +31.

33. The method of claim 30 wherein said mantissa table lookup uses two three-bit inputs and generates one four-bit output.

34. The method of claim 30 wherein said floating point format comprises only positive numbers.

35. A method of instantaneous queue congestion management of drop probabilities of packets using a queue congestion management algorithm applied to a plurality of queues sharing a common memory pool.

36. The method of claim 35 wherein said queue congestion management algorithm is implemented in hardware.

37. The method of claim 35 wherein said queue congestion management algorithm uses a floating point format.

38. The method of claim 37 wherein said floating point format comprises:

a four-bit normalized mantissa;

a six-bit biased exponent; and

performing multiply and divide operations with a mantissa table lookup.

39. The method of claim 38 wherein the most significant bit of said mantissa is implied.

40. The method of claim 38 wherein said biased exponent is in a range of -32 to +31.

41. The method of claim 38 wherein said mantissa table lookup uses two three-bit inputs and generates one four-bit output.

42. The method of claim 38 wherein said floating point format comprises only positive numbers.

43. The method of claim 35 wherein said queue congestion management algorithm comprises:

determining the total amount of shared memory space in bytes;

monitoring the instantaneous actual sizes of each of said queues;

dynamically calculating minimum and maximum queue sizes of a drop probability curve for each of said queues;

comparing said instantaneous actual sizes with said maximum and minimum queue sizes;

if said instantaneous queue size is less than said minimum queue size, then assigning said packet to said queue; otherwise

if said instantaneous queue size is between said maximum and said minimum queue sizes, then calculating and applying a drop probability using the slope of said drop probability curve; and otherwise

if said instantaneous queue size is greater than said maximum queue size, then dropping said packet.

44. The method of claim 35 wherein non-utilized shared memory space is allocated simultaneously to all of said queues sharing said common memory pool.

45. A method of time averaged congestion management of drop probabilities of packets using a weighted random early discard (WRED) algorithm applied to a plurality of arrays sharing a common memory pool.

46. The method of claim 45 wherein said arrays are queues, each of said queues being assigned to a different quality of service (QOS) priority level.

47. The method of claim 45 wherein said WRED algorithm is implemented in hardware.

48. The method of claim 45 wherein said WRED algorithm uses a floating point format.

49. The method of claim 48 wherein said floating point format comprises:

a four-bit normalized mantissa;

a six-bit biased exponent; and

performing multiply and divide operations with a mantissa table lookup.

50. The method of claim 49 wherein the most significant bit of said mantissa is implied.

51. The method of claim 49 wherein said biased exponent is in a range from -32 to +31.

52. The method of claim 49 wherein said mantissa table lookup uses two three-bit inputs and generates one four-bit output.

53. The method of claim 49 wherein said floating point format comprises only positive numbers.